An Auction and Witness Enhanced Trustworthy SLA Model for Decentralized Cloud Marketplaces

Zeshun Shi
z.shi2@uva.nl
University of Amsterdam
Amsterdam, the Netherlands

Siamak Farshidi
s.farshidi@uva.nl
University of Amsterdam
Amsterdam, the Netherlands

Huan Zhou
huanzhou@nudt.edu.cn
National University of Defense Technology
Changsha, China

Zhiming Zhao
z.zhao@uva.nl
University of Amsterdam
Amsterdam, the Netherlands

ABSTRACT

Cloud computing has become one of the most important technologies that have changed the traditional application development and operation (DevOps) lifecycle. However, current cloud software DevOps often faces the following key challenges: 1) selecting the best fitting service providers, customizing services and planning capacities for large-scale distributed applications; 2) guaranteeing high-quality and trustworthy service level agreements (SLAs) among multiple service providers; 3) enhancing the interoperability of cloud services across providers; 4) designing incentive model effectively among players. In this study, a framework called AWESOME is proposed to build a decentralized cloud marketplace and to address the above challenges. The proposed framework contains four subsystems including a customizable auction model, an incentive witness mechanism, and a social behavior-based simulator as one automated framework. We also provide a proof of concept to demonstrate that the AWESOME framework is feasible.

CCS CONCEPTS

• **Social and professional topics** → *Online auctions policy*; • **Security and privacy** → *Trust frameworks*.

KEYWORDS

decentralized cloud marketplace, auction, service level agreement

1 INTRODUCTION

Cloud computing paradigm provides flexible services based on pay-as-you-go business models [2]. Several well-known service providers maintain the traditional cloud marketplace, and the share of these top providers is continuously growing. According to the reports, as of October 2020, AWS, Azure, Google, and Alibaba control 63% of the entire cloud marketplace, whereas all other providers only share 37%[1]. Since product migration is complex, consumers become locked in a particular provider's ecosystems. In the future, however, we should expect a more open, fair, and trustworthy cloud resource trading marketplace for all service providers and customers.

There are usually two approaches to build a cloud marketplace: centralized and decentralized [8]. In a centralized cloud environment, all service trading transactions and trust-related issues rely on trusted third parties (TTP), e.g., some well-known cloud service providers with good reputations. However, those providers are not always trustworthy in practice and can be biased or conspire with any party. In a decentralized environment, however, transaction management and operations are performed by all sellers/buyers, which avoids the concentration of power and makes the transactions more trustworthy. In this case, all trust assurance comes from a distributed infrastructure (e.g., blockchain), which needs to be appropriately designed, implemented, deployed, and monitored.

Traditionally, Service Level Agreement (SLA) is a business concept that defines the contractual financial agreements between the roles who are engaging in the business activity. In the context of a cloud marketplace, it is an agreement between the cloud customer and provider regarding the cloud service quality [6]. For instance, the IaaS (Infrastructure-as-a-Service) provider, Amazon Elastic Compute Cloud (EC2), claims that the availability of its data center is no less than 99%. If this number is not achieved, it will pay back 30% credits to its customers as a compensation. In practice, however, this agreement is hard to enforce in a fair and transparent manner; it is usually performed manually and dominated by giant providers in the traditional SLA management process.

Blockchain technology brings in a new hint for possible solutions to address these challenges [9]. It inspires the emergence of a new decentralized cloud marketplace that encourages greater inclusivity and participation from different service parties. It is foreseeable

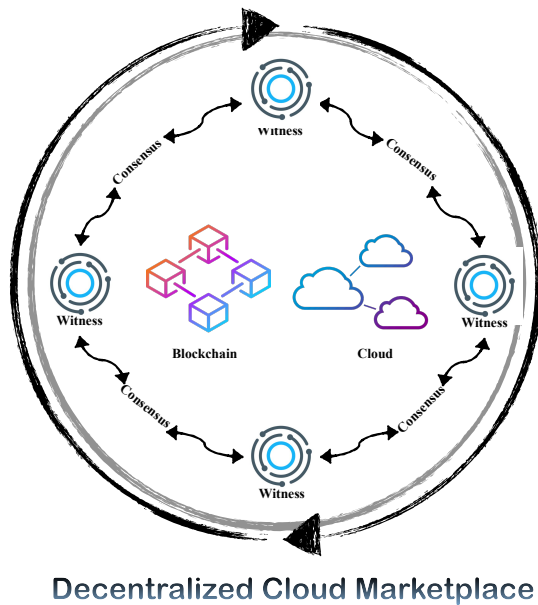---

[1]https://www.canalys.com/newsroom/worldwide-cloud-market-q320

Decentralized Cloud Marketplace

Figure 1: An Illustration of an Decentralized Cloud Marketplace.

that this decentralized marketplace will provide more choices and opportunities for both providers and consumers. Besides, the smart contract makes it possible to manage and automate the SLA process on the blockchain in a fair and tamper-proof way [7]. However, reaching a consensus on events that occur outside the blockchain is another possible issue. Cloud customers/providers can still violate the agreed SLA in a blockchain-based decentralized cloud marketplace. For example, the provider may not provide the QoS (Quality of Service) they promised, and the customer may refuse to pay for the claimed cloud resources. In the blockchain community, the bridge between on-chain and off-chain events is called "oracle" [5]. One of the solutions to build this bridge is to retrieve data from Oraclize[2], a third-party company performing as a trusted data source for the blockchain. However, this solution suffers from a single point of failure (SPOF) and needs extra commission fees. In this case, a decentralized and trustworthy witness mechanism is needed to judge SLA violations that occur off-chain.

This paper expects to enhance the trustworthiness of cloud auction and SLA by introducing a novel Auction and Witness Enhanced trustworthy SLA for Open, decentralized service MarkEtplaces (AWESOME) framework. Specifically, a new role called the witness is involved in the entire cloud service trading process, shown in Figure 1. The decentralized blockchain users can join the SLA judgment and work as witnesses through a carefully designed incentive mechanism that motivates witnesses to make effective judgments to win profits.

In the rest of this paper, we first demonstrate the current requirements and related works in building a decentralized cloud marketplace in Section 2. Two industrial use cases summarize these requirements and challenges. Then, to address these challenges, we

propose the AWESOME framework and present the detailed architecture overview and technical details in Section 3. Next, Section 4 shows a proof of concept to demonstrate the feasibility of the AWESOME framework. Finally, We conclude the entire paper in Section 5.

2 REQUIREMENTS AND RELATED WORK

In industrial innovations (e.g., crowd journalisms and smart transportation) and scientific applications (e.g., research data management and disaster early warning), cloud services are playing an increasingly important role in real-time processing information (e.g., multimedia acquired by mobile devices), running simulations (e.g., for predicting possible disasters) and for enabling extensive scale collaborations (e.g., for running distributed scientific workflows). Therefore, it is necessary to employ multiple data centers or providers to handle decentralized collaboration between data and resource providers and customers in several industrial use cases.

(1) Use case 1. Decentralized cloud marketplace for social media (taken from EU ARTICONF project): crowd journalism for real-time news reporting during live sports, music events, or natural disasters. Individual citizen journalists make photos or videos of the "news" and trade them via the news platform. The system has to detect fake news from those crowdsourced content by running real-time processing in a decentralized cloud service marketplace for those media contents, or engage human experts to review them.

(2) Use case 2. Decentralized service marketplace for medical data management (taken from EU CLARIFY project): sharing and utilizing pathology data provided by hospitals or individuals from different countries, where various medical data access constraints are often applied. When a machine learning application for studying breast cancer must use data from multiple hospitals, the application developer has to select cloud providers from a decentralized marketplace that meet the application needs (e.g., geolocation, capacity, and price).

We can therefore highlight the following requirements and challenges from those use cases:

- Provider selection, service customization, and capacity planning challenges. The developer has to select cloud services from different providers (very often multiple ones) due to distributed data locations (e.g., sensors or repositories), diverse data access constraints (e.g., for medical data), performance constraints (e.g., for real-time decisions in early warning). The various price and reputation models make the selection time-consuming and challenging to be optimal.
- SLA interoperability and guarantee challenge. The time-critical application constraints, e.g., for processing media contents during crowd news reporting and real-time decision-making, require the profound optimization of the application logic and components and the guarantee of the service quality of the cloud infrastructure, including both virtual machines and network connectivities. The diverse SLA terms among providers and the uncertainties in the SLA guarantee make performance optimization difficult.

___

[2]http://www.oraclize.it/

- Difficulties in verifying the incentive models and setting of the witness games in a decentralized marketplace. The business logic in a decentralized marketplace is often realized by smart contracts, which are supposed to be immutable after being deployed on blockchains. However, any careless design or mistake may cause unexpected loss.
- Virtual infrastructure automation challenge. When an application involves multi providers or data centers, the provisioning of the virtual infrastructure, deployment of the software platform and application components (often in terms of containers), monitoring, and adaptation of the application need to be ideally automated. However, the diverse Application Programming Interfaces (APIs) from different providers and the interoperability issues across those providers make the automated provisioning and deployment a challenge and result in high complexity for monitoring the runtime infrastructure quality and detecting SLA violations and adaptation of the infrastructure.

There are already many tools and academic studies focused on the challenges listed above. For example, Cloudsstorm [11] addresses DevOps resource management from the perspective of improving application programmability. BASIC [4] is an Ethereum blockchain-based urban agent simulator. It combines agent-based simulation with smart contract technology to verify the feasibility of using blockchain in simulated urban scenarios. By contrast, the authors in [10] and [1] use different auction models to achieve optimal cloud provider selection. While blockchain-based auction models have great potential for the cloud marketplace, most existing solutions focus only on the design of the auction models without considering the trusted execution of cloud SLAs. Although there are tools existing in different areas for different purposes (e.g., cloud/blockchain automation, DevOps, and blockchain simulation), there is no such a package solution to build a decentralized cloud marketplace and meet users' various application needs.

3 THE AWESOME FRAMEWORK

To tackle the current challenges in the decentralized cloud marketplace, we propose the AWESOME framework. The AWESOME software architecture consists of novel combinations of state-of-the-art technologies in DevOps, agent-based modeling, game theory, and blockchain. The proposal aims to tackle those challenges and to achieve the following objectives:

- Objective 1: Improve the provider selection in a decentralized ecosystem by developing an automated service auction framework to enable dynamic business relations between a consumer(s) and providers and establish an SLA.
- Objective 2: Improve the service quality and SLA's trustworthiness between consumer(s) and providers by establishing a decentralized dynamic witness mechanism to monitor the quality violations and automate the procedure for SLA compensation and payment.
- Objective 3: Improving the efficiency of smart contract validation by developing social behavior-based simulation components to evaluate and validate the impact of smart contracts for auctions and witnesses.

- Objective 4: Improve the continuous DevOps efficiency of an application in a decentralized cloud ecosystem by providing an integrated software framework to enhance the infrastructure management components in the existing DevOps stack. The AWESOME framework will be tested on both permissionless and permissioned blockchains.

3.1 Architecture Overview

The AWESOME framework consists of four subsystems in response to the proposed objectives, as shown in Figure 2:
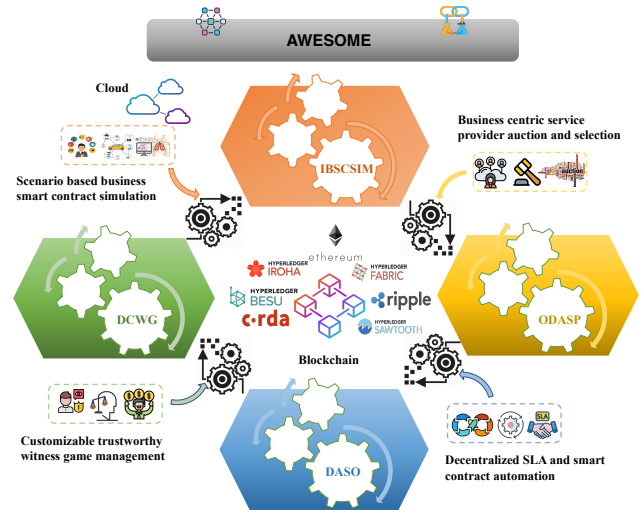
Figure 2: The Overall Architecture of the AWESOME Framework.

(1) The Interactive Business Scenario guided Smart Contract SIMulator (IBSCSIM) provides a smart contract business process simulation environment that connects both on-chain and off-chain activities. It aims to verify the feasibility of using blockchain in different use case scenarios (e.g., crowd journalism and pathology data sharing) through agent-based simulation by considering the communication among different smart contracts agents. More specifically, IBSCSIM designs simulation scenarios regarding 1) performance issues, 2) smart contract security issues, and 3) incentive model selection to provide users with overall DevOps guidance.

(2) On-Demand Auction for Service Providers (ODASP) provides an auction-based service provider selection solution. This subsystem will first diagnose the use case requirements and then select the most suitable auction model and algorithm to achieve the effectiveness of the auction process. The management of the auction process and the enforcement of the service fee payment (in the form of cryptocurrency) are all executed on the blockchain, ensuring that the whole auction is open and trustworthy. Finally, ODASP also audits bidder candidates to ensure that malicious providers cannot join the auction process.

(3) Decentralized Customizable Witness Game (DCWG) provides a game theory-based incentive framework to manage
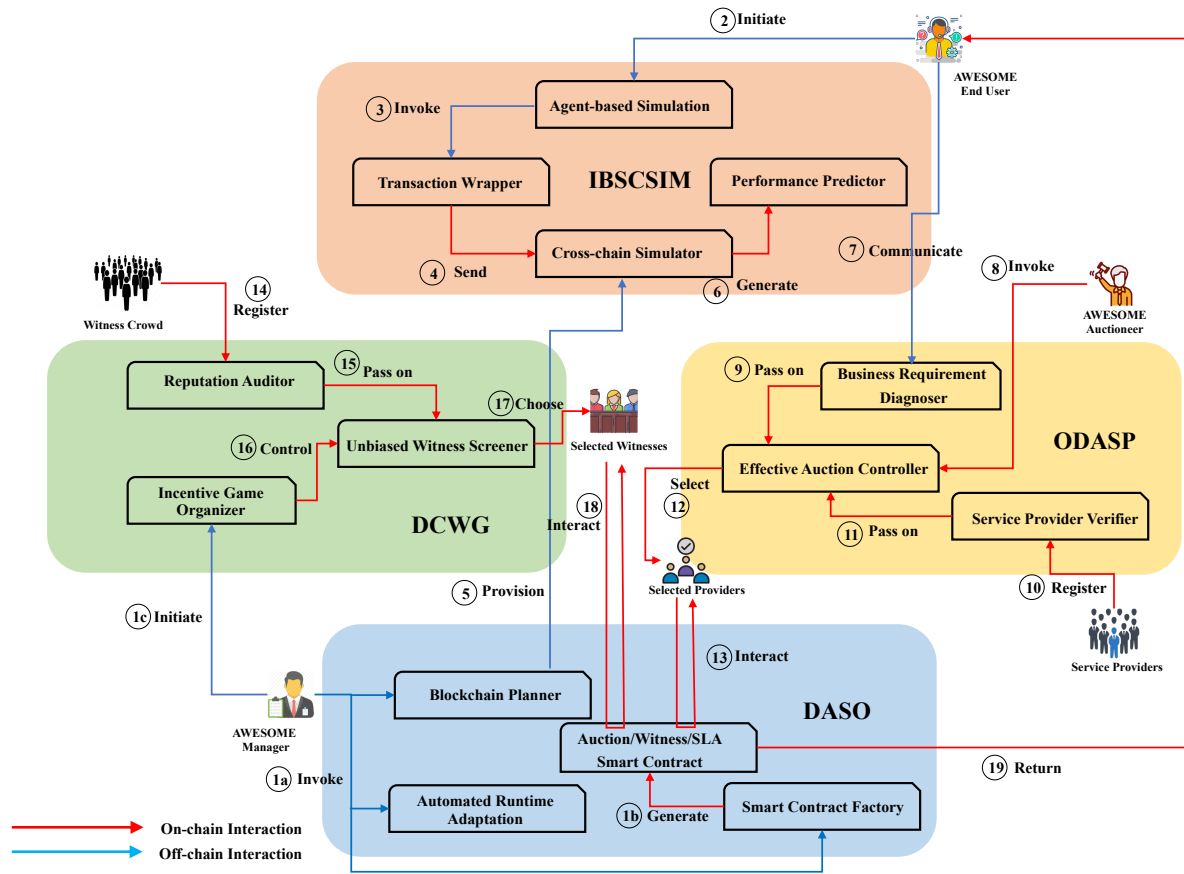
Figure 3: The Process Flow of the Awesome Framework and the Related Stakeholders in the Decentralized Service Ecosystem.

decentralized auction witnesses. First of all, an appropriate number of witness candidates will be selected in an unbiased way to perform off-chain monitoring of federated Cloud SLAs. DCWG will then design a game theory incentive mechanism (e.g., different payoff functions) to enable selected witnesses to make correct judgments to win more profits. The subsystem will also audit the witnesses' reputations to reward/restrict their participation in future monitoring activities.

(4) Decentralized Automated Service Orchestration (DASO) provides tools and APIs for application developers to set the necessary blockchain infrastructures. More specifically, it is responsible for automating the process of planning, provisioning blockchain infrastructure, and the generation and deployment of business/SLA smart contracts. The DASO subsystem also monitors and diagnoses smart contracts and the underlying blockchain infrastructure at runtime to provide effective adaptation.

3.2 Technical Details

As shown in Figure 3, the overall workflow of the AWESOME framework can be described as the following steps. First of all, a

AWESOME manager calls the DASO subsystem to plan and provision the blockchain infrastructure required for the simulation. The *AWESOME end user* then calls the IBSCSIM subsystem to initiate an agent-based on-chain simulation for the current use case. After that, IBSCSIM starts to simulate offline behaviors and generate on-chain predictions, which provides users with guidance on auction and witness settings in ODASP and DCWG. Meanwhile, the DASO subsystem automatically generates auction, witness, and SLA smart contracts to ensure trustworthy interaction between different participants. Next, the decentralized *service providers* and *witnesses* are registered in the ODASP and DCWG subsystems, respectively. When there are enough registered candidates, the *AWESOME manager* and the *AWESOME auctioneer* start the auction screening process to find qualified providers and witnesses. Finally, the *selected providers* collaborate to provide federated cloud services, and *witnesses* start to monitor the SLA to win profits. When the cloud service ends, the service price and witness fee will be paid and enforced with cryptocurrency using blockchain.

In the entire AWESOME workflow, IBSCSIM guides the selection of auction and witness game models and the on-chain deployment of the use case. ODASP selects candidate providers through an

effective auction mechanism. DCWG ensures trustworthy SLA enforcement through truth-telling witness monitoring. DASO provides automated infrastructure support for the entire process. The four subsystems form a dynamic ecosystem that provides services to AWESOME users in a collaborative way.

4 PROOF OF CONCEPT VALIDATION

In this section, we present the proof of concept validation of the AWESOME framework to demonstrate its feasibility. To meet the requirements of building a decentralized cloud marketplace, we designed three smart contracts using Ethereum blockchain (i.e., auction contract, witness contract, and SLA contract) to support trustworthy and fair interactions between different stakeholders[3]. In the AWESOME framework, we leverage a *smart contract factory* to manage and generate subcontracts instead of developing different contracts separately, as this is a more secure and efficient way [3].

4.1 Smart Contract Interactions

The sequence diagram below shows the interaction between the *contract factory* and the different subcontracts. First, an *AWESOME manager* calls the *contract factory* to create a new *auction contract*. Next, an *auction contract* with a customized auction rule for the business requirements is built to support transparency and automation of the auction process. In this case, decentralized service providers can register and submit their bids for services using commitments on the blockchain. The *auction contract* then selects the winning

Figure 4: The Sequence Diagram of the AWESOME Smart Contracts Interactions.

---

[3]AWESOME aims to build decentralized cloud marketplace using both permissionless and permissioned blockchains. We leave the implementation of other blockchain platforms (e.g., Hyperledger Fabric) as our future work.

providers based on the highest k bids and generates k *SLA contracts* for each provider[4].

When the auction is settled (note that the services have not been delivered yet), the *AWESOME manager* calls the *contract factory* again to generate a *witness contract* that contains customized incentive mechanisms to encourage truth-telling witnesses. More details about a game theory based witness payoff design are mentioned in our previous research [12]. Then, different winner providers can start to deliver cloud services off-chain, while the witnesses start to monitor all the services; if the QoS satisfies the requirements in the *SLA contract*, there is no violation; otherwise, there is a violation. The result of service monitoring is also returned to the *auction contract* and determines the status of the auction.

4.2 Cost Analysis

Regarding the implementation details of AWESOME contracts, we first designed some function interfaces in each smart contract, as shown in Table 1. These functions have built-in access control mechanisms; only specific stakeholder groups can access and call them. Next, we measured the transaction fee (ether) of each function interface, as shown in Figure 5. Specifically, three transaction submission modes (i.e., Low, Average, and High) were tested[5]. By analyzing the testing results, we can find that the transaction fees of most function interfaces are maintained at a relatively low level (less than 0.01 ether), except for only three special cases, namely *Place Bids* (auction contract), *Generate SLA Contracts* (auction contract), and *Calculate Witness Fee* (witness contract). These function

Table 1: The Gas Consumption of Each Function Interface in AWESOME Smart Contracts

Smart Contract	Access Control	Function Interface	Gas Consumption
Auction Contract	Customer	Setup Auction	171707
	Provider	Bidder Register	87310
	Provider	Submit Bids	134473
	Provider	Reveal Bids	110190
	Customer	Place Bids	435951
	Customer	Withdraw Deposit	44185
	Customer	Generate SLA Contracts	997266
Witness Contract	Witness	Witness Register	172679
	Witness	Submit Reports	110992
	Witness	Reveal Reports	150549
	Customer	Calculate Witness Fee	1108723
	Witness	Withdraw Witness Fee	22468
	Customer	Check Auction Settled	49571
SLA Contract	Provider	Set Customer	44074
	Provider	Set Service Duration	27266
	Provider	Publish Service	34587
	Provider	Setup SLA	55275
	Customer	Accept SLA	71536
	Provider	Cancel SLA	26202

---

[4]Note that in reality, different auction models have different rules. In this section, a reverse first-price sealed-bid auction (FPSBA) model is used as an example to illustrate the AWESOME framework workflow. Other auction models can be easily integrated on the smart contract

[5]The estimated transaction confirmation duration for three modes are 16 minutes, 2 minutes and 19 seconds, and 30 seconds, respectively. Data was collected on April 30, 2021 at https://etherscan.io/gastracker
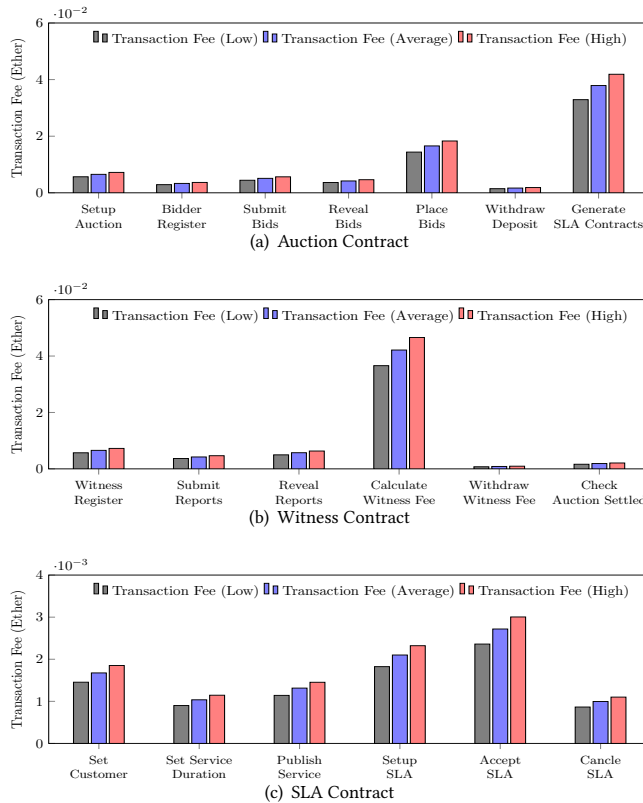
Figure 5: The Transaction Fee of Each Function Interface in AWESOME Smart Contracts

interfaces require certain computational tasks on-chain and therefore require more transaction fees than others. Nevertheless, we believe that the transaction involved in the three contracts is still very economical compared to the huge commission fee in the case of traditional auction houses that do not rely on the blockchain. All the above experiments prove that the smart contracts design of the AWESOME framework is feasible using blockchain infrastructure.

5 CONCLUSION

This work proposes a novel AWESOME framework for building a decentralized cloud marketplace, which is enhanced by auction models using witnesses. More specifically, the framework aims to reach the following objectives: 1) improve the efficiency of selecting service providers from a decentralized marketplace by proposing a decentralized auction framework; 2) enhance the trustworthiness of SLA among providers and users by using a dynamic decentralized witness model; 3) improve the verification efficiency of the smart contract in the service marketplace by using an interactive business simulator; 4) improve the continuous DevOps efficiency of the decentralized marketplace by integrating the AWESOME tool with the DevOps framework. We also provide a proof of concept validation, and the results demonstrate that the smart contract design solution of the AWESOME framework is feasible.

It is worth noticing that the AWESOME framework aims to develop highly modular software architecture for a decentralized cloud ecosystem. Some subsystem features (e.g., the cross-chain simulator in IBSCSIM and blockchain planner in DSLA) are still under development. We leave this development job as part of our future work. In the future, we will continue to test our framework and demonstrate its feasibility in two ongoing industrial projects (i.e., EU ARTICONF and CLARIFY).

ACKNOWLEDGMENTS

REFERENCES

[1] Marco Anisetti, Claudio A Ardagna, Piero A Bonatti, Ernesto Damiani, Marco Faella, Clemente Galdi, and Luigi Sauro. 2014. e-Auctions for multi-cloud service provisioning. In *2014 IEEE International Conference on Services Computing*. IEEE, 35–42.
[2] Shadi Ibrahim, Bingsheng He, and Hai Jin. 2011. Towards pay-as-you-consume cloud computing. In *2011 IEEE International Conference on Services Computing*. IEEE, 370–377.
[3] Yue Liu, Qinghua Lu, Xiwei Xu, Liming Zhu, and Haonan Yao. 2018. Applying design patterns in smart contracts. In *International Conference on Blockchain*. Springer, 92–106.
[4] Luana Marrocco, Eduardo Castelló Ferrer, Antonio Bucchiarone, Arnaud Grignard, Luis Alonso, Kent Larson, et al. 2018. Basic: Towards a blockchained agent-based simulator for cities. In *International Workshop on Massively Multiagent Systems*. Springer, 144–162.
[5] Roman Mühlberger, Stefan Bachhofner, Eduardo Castelló Ferrer, Claudio Di Ciccio, Ingo Weber, Maximilian Wöhrer, and Uwe Zdun. 2020. Foundational Oracle Patterns: Connecting Blockchain to the Off-chain World. In *International Conference on Business Process Management*. Springer, 35–51.
[6] Pankesh Patel, Ajith H Ranabahu, and Amit P Sheth. 2009. Service level agreement in cloud computing. (2009). https://corescholar.libraries.wright.edu/knoesis/78/
[7] Eder J Scheid, Bruno B Rodrigues, Lisandro Z Granville, and Burkhard Stiller. 2019. Enabling dynamic sla compensation using blockchain-based smart contracts. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 53–61.
[8] Rafael Brundo Uriarte, Huan Zhou, Kyriakos Kritikos, Zeshun Shi, Zhiming Zhao, and Rocco De Nicola. 2020. Distributed service-level agreement management with smart contracts and blockchain. *Concurr. Comput.-Pract. Exp.* (2020), e5800.
[9] Karl Wüst and Arthur Gervais. 2018. Do you need a blockchain?. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 45–54.
[10] Heli Zhang, Fengxian Guo, Hong Ji, and Chunsheng Zhu. 2017. Combinational auction-based service provider selection in mobile edge computing networks. *IEEE Access* 5 (2017), 13455–13464.
[11] Huan Zhou, Yang Hu, Xue Ouyang, Jinshu Su, Spiros Koulouzis, Cees de Laat, and Zhiming Zhao. 2019. CloudsStorm: a framework for seamlessly programming and controlling virtual infrastructure functions during the DevOps lifecycle of cloud applications. *Softw.-Pract. Exp.* 49, 10 (2019), 1421–1447.
[12] Huan Zhou, Xue Ouyang, Zhijie Ren, Jinshu Su, Cees de Laat, and Zhiming Zhao. 2019. A blockchain based witness model for trustworthy cloud service level agreement enforcement. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 1567–1575.